INTER PERSON VOICE CONVERSION USING FACTOR ANALYSIS

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Raghuram. A

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY,KANPUR

May 2005

CERTIFICATE

It is certified that the work contained in the thesis entitled "*Inter Person Voice Conversion Using Factor Analysis*" by *Raghuram A* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

GCRay 19.5.05

(G.C Ray)

May 2005

Professor,

Department of Electrical Engineering,

Indian Institute of Technology,

Kanpur-208016.

ABSTRACT

Voice Conversion is defined as modifying the speech signal of one speaker (source speaker) so that it sounds as if it had been pronounced by a different speaker (target speaker). In this thesis, we present a method for voice conversion by representing the joint probabilistic acoustic space of the two speakers with a Mixture of Factor Analyzers (MFAs). This can also be interpreted as a reduced dimension mixture of Gaussians.

Most of the existing voice conversion systems are trained on aligned LSF vectors. However, there are many applications of voice conversion systems where the amount of training data from the source speaker and the target speaker is different. The amount of source data is large, but it is desired to estimate the transformation with a small amount of target data. The extra unaligned source data is incorporated into the training phase to estimate the parameters of the MFA and hence improve performance.

Objective experiments demonstrate that the performance of the proposed system using factor analyzers is comparable to the performance obtained using existing systems using Gaussian mixture models, with significant gains in both time and memory complexity. The addition of unaligned data in the training phase leads to a much superior performance in conversion. Subjective tests imply that small increments in the dimension of the factor analyzers does not make a difference perceptually to the listener when the increments are small.

To My Parents

Acknowledgements

I am grateful to Prof. G.C Ray for giving me an opportunity to work on the problem of voice conversion and for the support and motivation throughout the course of the thesis. In addition I also thank the faculty and the Department of Electrical Engineering, IIT Kanpur in general for the help and resources throughout the course of my master's program.

To my colleagues, Dilip, Hari, Hemanth, Jawahar, Mayank, Pulkit, Rajesh and Vijay, thanks for your wonderful company. And to everyone who has participated in the subjective tests: Thanks for your patience and time.

I am indebted to my sister Rashmi and my brother-in-law Sonjai for their love and encouragement throughout the course of my thesis. Most importantly, I must thank my parents for making me who I am, for always trusting me in making the right decision and for supporting me in whatever I do.

Contents

List of Figures

Chapter 1

Introduction

Speech Signals convey a wide range of information. Among them, the meaning of the message being uttered is of prime importance. However, secondary information such as speaker identity also plays an important part in oral communication. Voice modification techniques attempt to transform the speech signals uttered by a given speaker so as to alter the characteristics of his or her voice. This problem - how to modify the speech of one speaker so that it sounds as if it was uttered by another speaker - is generally known as voice conversion (VC) [1].

In daily life, the individuality of voices allows us to recognize between different speakers. Also speaker identity makes it possible to differentiate between speakers in a conference call or on a radio program. Consequently there are a number of useful applications for controlling the speaker identity by means of a VC system, especially when integrated into other speech systems with either synthetic or natural speech output.

An example application is the integration of a VC system with a text-to-speech (TTS) synthesizer. Today's state-of-the-art TTS systems are based on a concatenative synthesis method in which a system retrieves natural speech segments from a database and joins them together to generate a new utterance. The synthesis database contains an organized collection of carefully recorded speech, and the speaker identity of the synthesis output bears resemblance to the original speaker identity of the database speaker. The creation of a synthesis database for a new synthesis voice is a significant recording and labelling effort, and requires significant amount of computational resources.

Using VC technology, new synthesis voices can be created by novice users quickly and inexpensively by creating a "speaker model" from a small number of speech utterances produced by the desired target speaker. The speaker model describes the characteristics of the target speaker's voice. Using different speaker models, the synthesis system can generate speech signals with different speaker identities from a single speaker database, which plays the role of the source speaker [2, 3, 4].

Another application is in the area of very-low-bandwidth coding of speech. Speech coding systems that are designed to operate at 2400 bps or less do not preserve speaker identity during transmission [6]. For these systems, VC algorithms have the potential to render the decoded speech at the receiver so that it matches the speaker identity of the transmitting speaker.

Voice Conversion systems can also be used in language interpreters and cross lan-

guage voice conversion [7]. Researchers have also considered a VC system for rendering acoustically impaired speech more intelligible [10, 9].

1.1 General Voice Conversion System

In voice conversion, we map the acoustic features of a *source* speaker to those of a *target* speaker. Figure 1.1 illustrates the typical model for a voice conversion system.

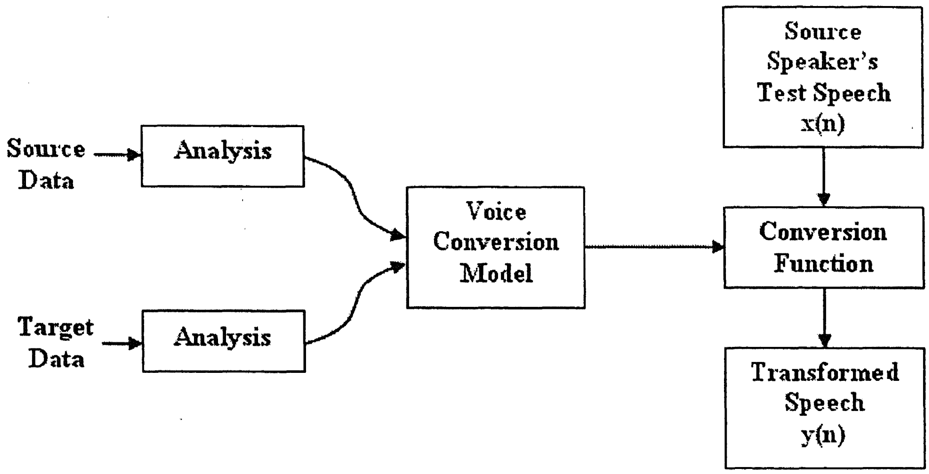

Figure 1.1: General Voice Conversion System

We collect speech in a parallel training corpus from both the source and target speaker for use in training the model. After training is complete, we predict what a target speaker sounds like using the information from the new speech of the source speaker.

A typical voice conversion system consists of seven components: the speech corpus, time alignment of the speech, analysis, training, voice mapping, synthesis, and post-processing.

Speech corpus -We must collect the same speech from a set of speakers. This speech should contain an even distribution of the phonemes of the speakers language so that we can model a variety of sounds; this variety improves the quality of the voice conversion system. In this work, we use the Arctic Corpus for training and testing [11].

Alignment of phonemes -We should have a robust method for time aligning the source and target speakers speech. Exact time alignment before training is necessary for optimal performance of this particular voice conversion system.

Analysis -We must determine the relevant acoustic features to use in training the system. These features should be able to represent a large portion of speech with only a few parameters. We discuss the features we use for representing speech in §2.3 and §2.5. Examples of typical features are the pitch or energy of a short segment of speech.

Training -We must figure out an appropriate model for training the voice conversion system.

Voice Mapping -We perform the mapping of the source speakers features to the target speakers. Usually, this mapping is a statistical expectation. Training and mapping are the crucial items that we focus on in this thesis while also being the subject of most research. We mention a few of the previously employed techniques in §2.7.

Synthesis -We must synthesize the transformed features into high quality speech.

Post-processing -After synthesis, we might perform some final processing of the signal to improve its quality.

1.2 Thesis Outline

The following chapters are summarized below:

Chapter 2 introduces the mathematical model for human speech production and presents the features with which we represent speech.

Chapter 3 provides an introduction to a previous model for voice conversion. We then discuss the present research, a method of modelling the probabilistic acoustic space of both speakers with a Mixture of Factor Analyzers and performing conversion with this model. We also discuss a method to improve the performance of the voice conversion system by using unaligned data.

Chapter 4 presents the objective and subjective results of our voice conversion system. It highlights the tradeoff of performance versus complexity that our system offers.

Chapter 5 concludes and also discusses future opportunities for research in voice conversion.

Chapter 2

Theoretical Background

In order to develop an effective voice conversion system it is important to understand the fundamental properties of speech. This chapter provides the background on how speech is produced and how their acoustics are modelled mathematically, the different speaker characteristics and a brief review on the previous methods for voice conversion.

2.1 Physiology of Speech Production

The main speech organs of the human speech production system is shown in Figure 2.1. Speech is produced by a part of the human anatomy called the vocal tract, which begins at the vocal cords, or *glottis*, and ends at the lips.

Air enters the lungs via the normal breathing mechanism. As air is expelled from the lungs through the *trachea* (or windpipe), the tensed vocal cords within the *larynx* are caused to vibrate by the air flow. The air flow is chopped into quasi-periodic pulses which are then modulated in frequency in passing through the *pharynx* (the throat

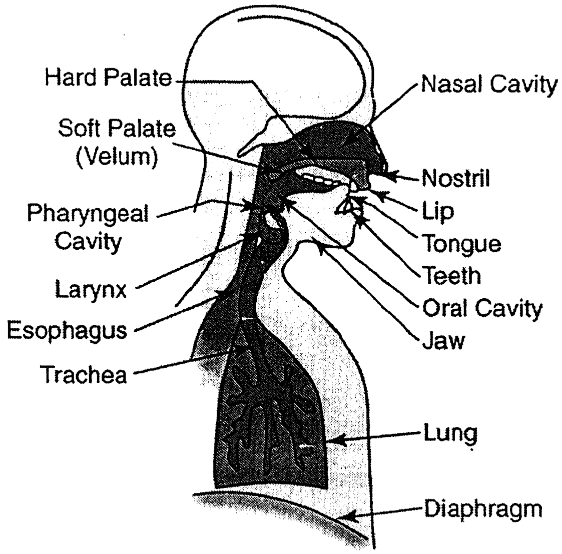

Figure 2.1: Human Speech Production

cavity), the mouth cavity, and possibly the nasal cavity. Depending on the positions of the various articulators (i.e., jaw, tongue, velum, lips, mouth), different sounds are produced. Consequently, the air flow is the source for four types of sounds [13, 12]:

Aspiration noise - The sound of air rushing through the entire vocal tract, similar to breathing through the mouth.

Frication noise - The sound of turbulent flow at a point of narrow constriction, for example during the initial sound in "fair".

Plosion - The sound of an initial burst, for example during the initial consonant in "ton".

Voicing - A quasi-periodic vibration of the vocal cords or *glottis*, for example during the vowel in "key". The frequency of vibration is called the *fundamental frequency* or

F_0 and is perceived as *pitch*.

The four types of sounds can occur in combination. For example, the initial sound in "vault" combines frication noise with voicing. Examples of voiced sounds in the English alphabet are the sounds produced when pronouncing the vowels *a*, *e*, *i*, *o* and *u*. Unvoiced sounds are produced by forcing air through the lungs and forming a constriction at some point in the vocal tract. Unvoiced utterances are noisy in nature; examples are the sounds produced when saying the English consonants *s* and *f*.

2.2 Speaker characteristics

The acoustic speech signal contains many types of information. Primarily, the signal carries information about the message (*what* was said), but also includes information about the speaker (*who* said it) and the environment (*where* it was said). Speaker characteristics describe the aspects of speech that are related to the person that produced it, independent of the message and the environment. The task of a voice conversion is thus to change the speaker characteristics of a speech signal, while preserving other types of information. The characteristics of a speech signal are commonly divided into the following types of cues:

Segmental cues - These describe the "sound" or "timbre" of the speaker's voice. Acoustic descriptors of segmental cues include formant locations and bandwidths, spectral tilt, F_0 and energy. Segmental cues depend mainly on the physiological and physical properties of the speech organs and the speaker's emotional state [15].

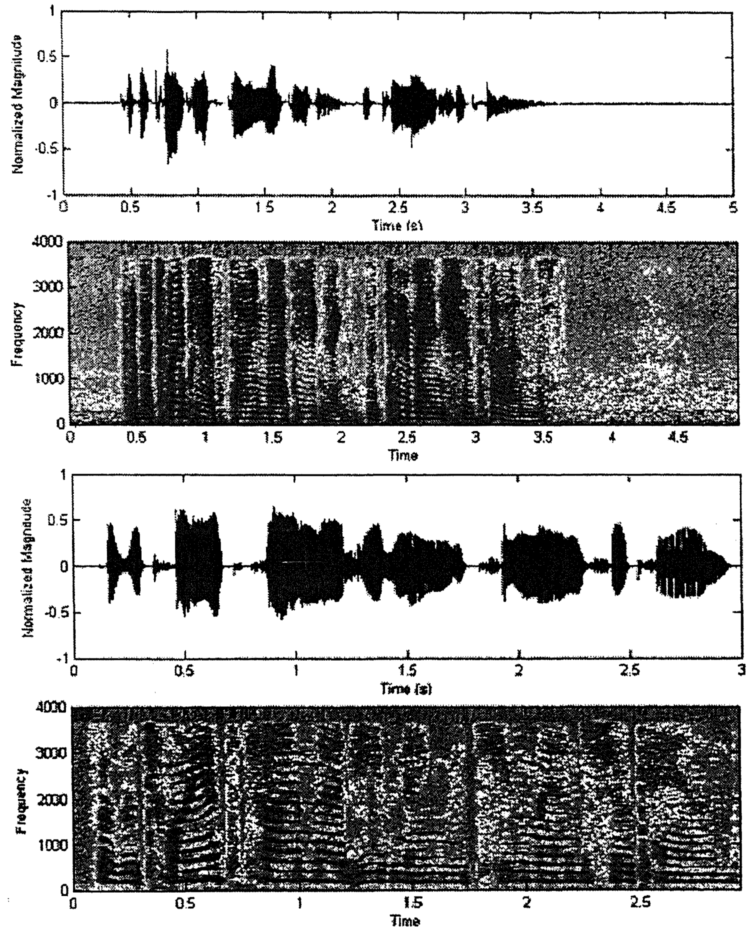

Figure 2.2: Speech Signals and Spectrograms for Two Speakers

Suprasegmental cues - These describe the prosodic features related to the style of speaking, for example the duration of phonemes and the evolution of F_0 (intonation) and energy (stress) over an utterance. The average behavior of phoneme duration, F_0, and energy are perceived as *rate of speech*, *average pitch*, and *loudness.* These cues are influenced by social and psychological conditions [14].

Linguistic cues - These include particular choices of words, dialects and accents.

Linguistic cues are beyond the scope of this thesis and will not be considered.

We will illustrate some of the segmental and suprasegmental cues by considering the differences between two different speakers in an example. Figure 2.2 shows the waveforms and spectrograms for two speakers uttering the sentence "For the twentieth time that evening, the two men shook hands". We see how the magnitude of the formant changes over time with respect to the different sounds uttered by the two speakers. One of the differences in suprasegmental cues are manifested in the different duration lengths of the same speech spoken by the different speakers. The Suprasegmental cues can be changed easily by the speaker. However, segmental cues are closely linked to the physiology of the speech production organs and can thus be considered as inmutable.

2.3 Acoustic Filter Model

Speech is a highly non-stationary process; i.e., the statistics of the underlying signal vary with respect to time. But, for short segments of time, speech is either quasi-periodic or noisy. So we can assume that speech is wide-sense stationary - the first and second order statistics remain constant during these short segments. Thus, for these short segments of time, we can form a tractable model to represent the speech production process as described in [16].

A Powerful method for modelling a discrete-time system for a short segment of time is the Linear Predictive Coding (LPC) model. The basic idea behind the LPC model is that a given speech sample at time n, $s(n)$, can be approximated as a linear

combination of the past p speech samples, such that

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + ... + a_p s(n-p) \tag{2.1}$$

where the coefficients $a_1, a_2, ..., a_p$ are assumed constant over the speech analysis frame. Equation 2.1 can be converted into an equality by including an excitation term, $Gu(n)$, giving:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n) \tag{2.2}$$

where $u(n)$ is a normalized excitation and G is the gain of the excitation. Equation 2.2 can be expressed in the z-domain as

$$S(z) = \sum_{i=1}^{p} a_i z^{-i} S(z) + GU(z) \tag{2.3}$$

leading to the transfer function

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \tag{2.4}$$

The normalized excitation source, $u(n)$ is scaled by the gain, G, and acts as the input to the all-pole system, $H(z) = \frac{1}{A(z)}$, to produce the speech signal, $s(n)$. Based on prior knowledge that the actual excitation function for speech is essentially a quasi-periodic pulse train (for voiced speech sounds) or a random noise source (for unvoiced sounds), the appropriate synthesis model for speech, corresponding to the LPC analysis, is as shown in Figure 2.3 [17].

This view of speech production is very powerful and it can explain the majority of speech phenomena. The normalized excitation source is chosen by a switch whose

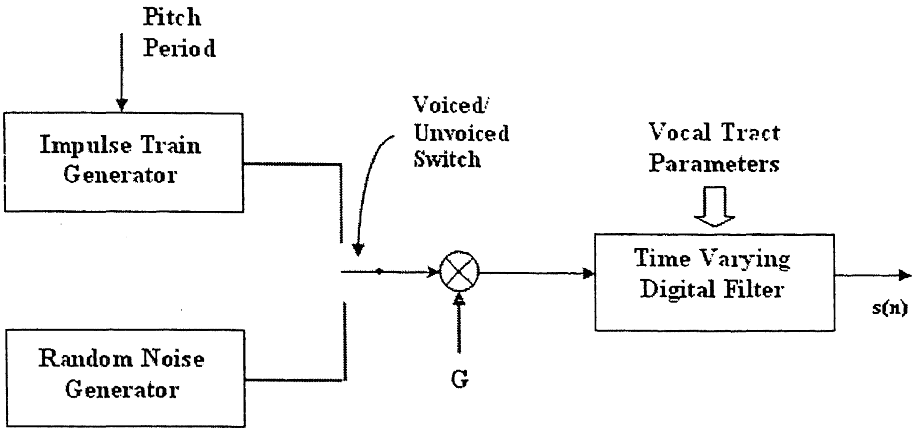

Figure 2.3: Speech synthesis model based on LPC model

position is controlled by the voiced/unvoiced character of the speech, which chooses either a quasi-periodic train of pulses as the excitation for voiced sounds, or a random noise sequence for unvoiced sounds. The appropriate gain G of the source is estimated from the speech signal, and the scaled source is used as input to the digital filter ($H(z)$), which is controlled by the vocal tract parameters characteristic of the speech being produced. The parameters of the model are thus voiced/unvoiced classification, pitch period for voiced sounds, the gain parameter, and the coefficients of the digital filter, $\{a_k\}$. The model described above has the advantage that the computation of the a_k coefficients is easily tractable with the Levinson's algorithm.

2.4 The Levinson-Durbin Algorithm

In this section we provide a brief description of the Levinson algorithm which he formulated in 1947 [18] with performance improvements made by Durbin [19] for the specific

problem of a time series as we have with our speech model in Equation 2.2. Levinson's algorithm runs in $O(p^2)$ time compared with older methods which run in $O(p^3)$ time.

In order to present the Levinson algorithm we first consider the linear combination of past speech samples as the estimate $\tilde{s}(n)$, defined as

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{2.5}$$

The prediction error $e(n)$ is defined as,

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.6}$$

To set up the equations that must be solved to determine the predictor coefficients, we define short term speech and error segments at time n as

$$s_n(m) = s(n+m) \tag{2.7}$$

$$e_n(m) = e(n+m) \tag{2.8}$$

and we seek to minimize the mean squared error signal at time n

$$E_n = \sum_m e_n^2(m) \tag{2.9}$$

which can be written as,

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right]^2 \tag{2.10}$$

To solve Equation 2.10 for the predictor coefficients, E_n is differentiated with respect to each a_k and set to zero,

$$\frac{\partial E_n}{\partial a_k} = 0, \quad k = 1, 2, \ldots p \tag{2.11}$$

yielding

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k) \tag{2.12}$$

Expression 2.12 can be written as

$$\phi_n(i,0) = \sum_{k=1}^{p} \hat{a}_k \phi_n(i,k) \tag{2.13}$$

which describes a set of p equations in p unknowns. It may be observed that the terms of the form $\sum_m s_n(m-i)s_n(m-k)$ are terms of the short-term covariance $s_n(m)$, i.e.

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k) \tag{2.14}$$

The minimum squared error, $\hat{E}_n$ can be expressed as

$$\hat{E}_n = \sum_m s_n^2(m) - \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m)s_n(m-k) \tag{2.15}$$

It can be shown under the assumption that the speech segment $s_n(m)$, is identically zero outside the range $0 \leq m \leq N-1$ that $\phi_n(i,k)$ defined before is identical to the short time autocorrelation function r_n.

$$\phi_n(i,k) = r_n(i-k) \tag{2.16}$$

where

$$r_n(k) = \sum_{m=0}^{N-1-k} s_n(m)s_n(m+k) \tag{2.17}$$

Since the autocorrelation function is symmetric, i.e. $r_n(-k) = r_n(k)$, the LPC equations can be expressed as

$$\sum_{k-1}^{p} r_n(|i-k|)\hat{a}_k = r_n(i), \;\; 1 \leq i \leq p \tag{2.18}$$

The optimal solution for $\{a_k\}$ is well known as the Wiener-Hopf solution, and we can use Durbin's method to solve for the coefficients recursively. The algorithm finds the p^{th} order solution for $\{a_k\}$ by using the $(p-1)^{th}$ order solution. We give a brief description of the algorithm below.

$$E^{(0)} = r(0) \tag{2.19}$$

$$\kappa_i = \{r(i) - \sum_{j=i}^{i-1} \alpha_j^{(i-1)} r(|i-j|)\}/E^{(i-1)}, \quad 1 \leq i \leq p \tag{2.20}$$

$$\alpha_i^{(i)} = \kappa_i \tag{2.21}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - \kappa_i \alpha_{i-j}^{(i-1)} \tag{2.22}$$

$$E^{(i)} = (1 - \kappa_i^2) E^{(i-1)} \tag{2.23}$$

The above equations are solved recursively for $i = 1, 2, ..., p$. The final solution is given as

$$a_m = LPC\ coefficients = \alpha_m^{(p)}, \quad 1 \leq m \leq p \tag{2.24}$$

The algorithm exploits the toeplitz nature of the covariance matrix thus leading to significant gains in computational efficiency.

2.5 Line Spectral Frequencies

Line spectral frequency (LSF) is an alternative representation of LPC [20]. LSFs have several desirable properties which are discussed in this section.

Two polynomials *P(z)* and *Q(z)* of order *p+1* are formed from the order p prediction

error filter $A(z)$ in the following manner.

$$P(z) = A(z)\left(1 + z^{-(p+1)}\frac{A(z^{-1})}{A(z)}\right) = A(z)(1 + G(z)) \tag{2.25}$$

$$Q(z) = A(z)\left(1 - z^{-(p+1)}\frac{A(z^{-1})}{A(z)}\right) = A(z)(1 - G(z)) \tag{2.26}$$

with

$$G(z) = z^{-(p+1)}\frac{A(z^{-1})}{A(z)} \tag{2.27}$$

Therefore $A(z)$ can be written as

$$A(z) = \frac{P(z) + Q(z)}{2} \tag{2.28}$$

The LSFs are defined as those values of frequency ω such that

$$\{\omega | P(e^{j\omega}) = 0 \; or \; Q(e^{j\omega}) = 0; \; 0 < \omega < \pi\} \tag{2.29}$$

Thus LSFs are frequency values associated to the unit magnitude zeros of $P(z)$ or $Q(z)$ [21]. The important properties of LSFs are enumerated below.

1. If $A(z)$ is minimum phase, then all zeros of $P(z)$ and Q(z) are on the unit circle.

2. The LSFs ω_p of $P(z)$ and the LSFs ω_q of $Q(z)$ are interleaved with one another, i.e.,

$$0 < \omega_{p1} < \omega_{q1} < \ldots < \omega_{pi} < \omega_{qi} < \ldots < \pi \tag{2.30}$$

   This interleaving property provides a easy way to verify the stability of the underlying synthesis filter.

3. The LSFs have good interpolation properties and quantize well because they are more evenly distributed than LPCs [22].

4. Two LSFs that are close in value correspond to a peak in the spectrum, and the peak can be interpreted as a formant [23]. LSFs can thus correlate well with the formant frequencies that identify a speaker.

These properties of the LSF are desirable for a voice conversion system and thus LSF is the chosen representation for the speech features.

2.6 Analysis

This section describes how the speech features are calculated from the speech signal and the various choice of parameters involved in the analysis phase.

We perform the analysis, processing and synthesis of speech by considering a small section of speech at a time. The original speech waveform is apportioned into small, overlapping frames $s^m(n)$, thus the system is said to be frame based.

Pitch and voiced-unvoiced decisions have to be estimated for each frame. Various methods have been reported in literature for estimating pitch and making voicing decisions in a frame. In this work we follow the method given by Ahmadi and Spanias [24]. The LPC filter coefficients a_k are calculated using the Levinson's algorithm described in §2.4.

Figure 2.4 shows the effect of LPC prediction order, p, on the RMS prediction error e_n, for both sections of voiced speech and unvoiced speech [26]. It is seen that

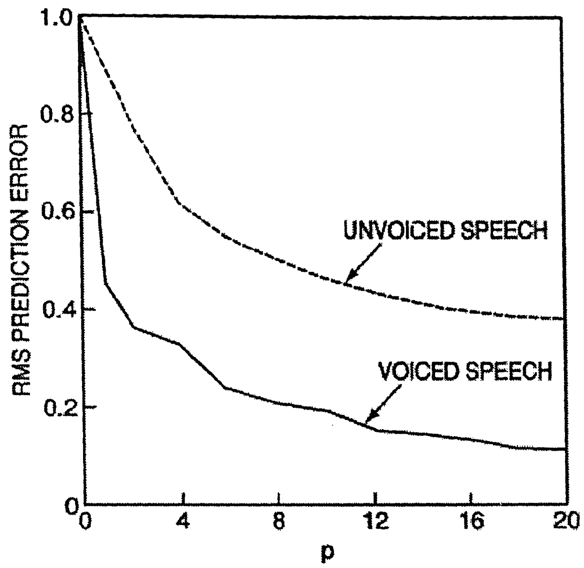

Figure 2.4: Variation of RMS prediction error with the number of prediction coefficients, p

the prediction error for unvoiced speech is more than that for voiced speech. The result is intuitive as unvoiced speech is less linearly predictable than voiced speech. For prediction orders greater than *12* the curve is relatively flat and results in a less parsimonious representation of the sound. We have therefore considered an prediction order of *16* is this work. The obtained set of LPCs $a_i, i = 1, ..., p$ are converted into the alternative LSF representation with the aid of a root finding procedure proposed in Soong and Juang [25].

2.7 Previous Methods for Voice Conversion

Most existing voice conversion systems employ the methodology above and that described in §1.1. In this section we give a brief review on previous methods for voice

conversion.

Codebook Mapping

One of the earliest approaches to the voice conversion problem is the mapping codebook approach of Abe *et al.* [8], which was originally introduced by Shikano *et al.* for speaker adaptation [28]. In Abe's approach, a clustering procedure - vector quantization (VQ) is applied to the spectral parameters of both the source and the target speakers. The two resulting VQ codebooks are used to obtain a mapping codebook whose entries represent the transformed spectral vectors corresponding to the centroids of the source speaker codebook. The main shortcoming of this method is the fact that the parameter space of the converted envelope is limited by a discrete set of envelopes, causing a drop in the quality of the converted speech. Arslan [30] extended this work by mapping not only the LSFs, but also the excitation; in addition, he improved the method by which the transformation weights were estimated. He named his method for updating the weights Codebook Weight Update by Gradient Descent; it is an optimization technique that iterates until the energy of the weight errors falls below a threshold. He integrated this method into his larger framework for voice conversion called Speaker Transformation Algorithm using Segmental Codebooks (STASC). Turk extended the STASC framework by integrating subband voice conversion using wavelets and selective pre-emphasis [31, 32]. Orphanidou [33] utilized the Generative Topographic Mapping [35] in reducing the dimensionality when mapping codebooks.

Mixture Modelling

Stylianou modelled the acoustic probability space of the source speaker with a Gaussian Mixture Model (GMM) in [36, 37]. He then found the cross-covariance of the source and target vectors and the mean target vector using least squares optimization of an overdetermined set of linear equations. In his work, he demonstrated the theoretical superiority of the GMM by showing that vector-quantization methods for voice conversion are a special case of the GMM in which only the mean of a cluster is mapped. Toda [38] implemented the GMM algorithm for voice conversion within their STRAIGHT analysis-synthesis framework. Kain extended Stylianou's work by modelling the joint probability density function of both the source and target speakers [5, 3]. This method obviates the need to perform the least squares optimization as with Stylianou's method. Modelling the joint probability density allows the system to capture all possible correlations between the source and target speakers spectrum. Various enhancements have also been proposed to Kain's method by Young [39]. Mark Wilde [34] utilized probabilistic principal component analysis to solve the problem. Recently, Mouchtaris has proposed a novel algorithm for non-parallel training algorithm for voice conversion by maximum likelihood constrained adaptation [40].

Other Methods for Spectral Conversion

In [47], Lee used an orthogonal vector space transformation to convert voiced speech and a non-linear neural net predictor to model the excitation; he converted the excitation using mapping codebooks. Ning Bi used linear multivariate regression for map-

ping [9]. Watanabe used radial basis functions for performing the spectral mapping [44], and Narendranath used neural networks [45] with success. Salor [61] employed a least mean square adaptive filtering technique to filter the target speakers features from the source speakers. Although mapping with codebooks may be less expensive computationally, this method is less robust than mixture modelling, as the reduction of a continuous spectral space into a discrete codebook introduces quantization noise leading to a degradation in the quality of the converted speech.

2.8 Thesis and Proposed Method

We extend the spectral mapping aspect of voice conversion by modelling the joint probability space of both speakers with a Mixture of Factor Analyzers (MFAs). Previous methods that used the GMM to model the space are constrained to only two possible selections for representing the covariance structure - diagonal and full covariance matrices. With diagonal structure, the training time is quick but conversion performance is sacrificed. With full covariances, we can model the underlying second order statistics with improved conversion performance but incur the penalty of longer training time.

By modelling covariance structure with a Mixture of Factor Analyzers, we provide an entire range of covariance structure for the user to manipulate depending on the quality of synthesized speech. The existing systems for voice conversion require time aligned data of the source speaker and the target speaker. We also discuss a method to improve the performance of the voice conversion system using Mixtures of Factor

Analyzers by including extra unaligned data in the training phase. Objective and subjective tests are then presented by evaluating the proposed method against the system using GMM.

Chapter 3

Transforming the Spectral Envelope

The objective of spectral transformation is to find a statistical mapping between those features of the source and target speakers which best represent their vocal tracts. After finding this mapping, speech is synthesized using the transformed features.

In this chapter, we first discuss some pre-processing steps taken before training. We then highlight the baseline system which has been implemented for representing the probabilistic acoustic space of both speakers followed by the method for voice conversion. This system is designed to transform the spectral envelope of speech by changing parameters of an all-pole model, using a transformation function implemented by a Gaussian mixture regression model. In the later part of the chapter, we discuss our model which employs factor analysis that provides a more parsimonious model of the space with a number of advantages. The conversion function for the case when a mixture of factor analyzers is used, is derived. We also discuss the use of extra unaligned data in the algorithm to improve the performance.

3.1 Time Alignment

In our frame based system, the features of one frame describe only a small portion of speech and thus a sequence of features, or *feature stream*, represents an entire utterance. Because of natural variations in the durations of linguistic units between different speakers, the feature streams of the source and the target speaker must be time aligned before training the voice conversion model. We use the common method of dynamic time warping (DTW) to align the waveforms [17].

First, we trim both speakers waveforms with an algorithm that removes silence both at the beginning and the end so that the DTW algorithm has a better initial alignment [17]. The goal of time-alignment is to modify the source and target speech

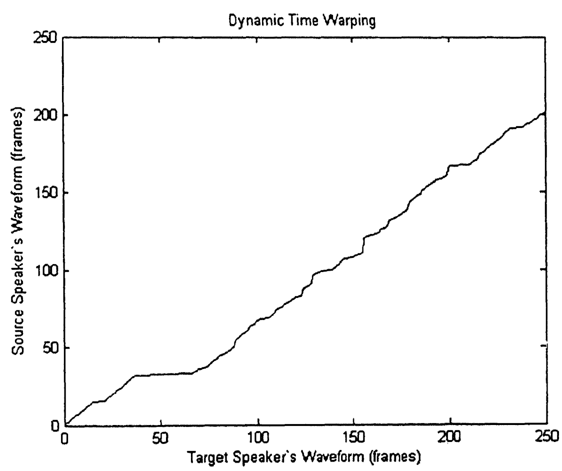

Figure 3.1: Example of an alignment path

LSF feature stream in such a way that the resulting feature streams can be thought of as describing the same phonetic content frame by frame. We achieve alignment

by selectively deleting or repeating frames from the target speaker feature stream to match the number of source frames. Alternatively, we can avoid deleting any frames altogether by stretching the shorter region of one speaker to the length of the longer one of the other speaker. DTW uses a dynamic programming strategy to find this optimal path. An example of the alignment path is shown in Figure 3.1. After alignment, we collect aligned LSF feature vectors into N frames of source data

$$X_{p\times N} = [L^1_{source}\ L^2_{source}\ \dots\ L^N_{source}\ ] \tag{3.1}$$

and respectively, target data

$$Y_{p\times N} = [L^1_{target}\ L^2_{target}\ \dots\ L^N_{target}\ ] \tag{3.2}$$

Beginning and ending silences are not included in the training data sets. An example

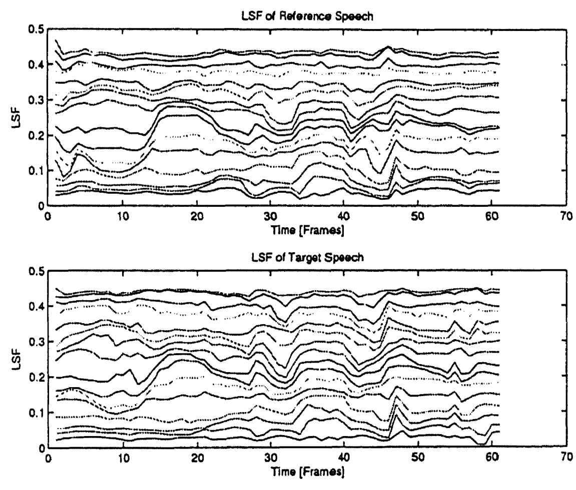

Figure 3.2: Two aligned LSF feature streams

of a single sentence of two aligned LSF feature streams is shown in Figure 3.2. The value of N depends on the amount of training data.

3.2 Training

After feature extraction and time alignment, we model the joint probability space of the source and target vectors **x** and **y** for all N frames of speech. The purpose of the training stage is to estimate parameters of a transformation function so that it can predict target speaker features **y** from the source speaker features **x**. We predict the best estimate of the target vector **y** given the source vector **x** with a conditional expectation $E[\mathbf{y}|\mathbf{x}]$.

3.2.1 The Gaussian Mixture Model

For determining the joint probability density $p(\mathbf{x}, \mathbf{y})$, we consider the concatenation of the source and target feature vectors as the d-dimensional vector **z** for ease of notation.

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{3.3}$$

A mixture model allows the probability distribution of **z** to be modelled as a weighted sum or mixture of M component densities, also referred to as classes [41]. This is given by Equation 3.4.

$$P(\mathbf{z}) = \sum_{j=1}^{M} p(\mathbf{z}/j)P(j) \tag{3.4}$$

Each $P(j)$ is a mixing weight or prior probability of component j occurring. The mixing weight satisfies the constraint that $\sum_{j=1}^{M} P(j) = 1$. In the case of a Gaussian mixture model, each component density is a d-variate Gaussian function of the form

$$p(\mathbf{z}/j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} exp\left\{-\frac{1}{2}(\mathbf{z} - \mu_j)'\Sigma_j^{-1}(\mathbf{z} - \mu_j)\right\} \tag{3.5}$$

where $\boldsymbol{\mu}_j$ and Σ_j denote the mean and covariance of the j^{th} component of the mixture model.

The complete Gaussian mixture density is parameterized by the mean vectors, the covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation

$$\lambda = \{P(j), \boldsymbol{\mu}_j, \Sigma_j\}, \quad j = 1, ..., M \tag{3.6}$$

Rather than model the joint probability space of the speakers with a discrete set of vectors as do codebook techniques [8, 30], the GMM models this space as a continuous probability density. The benefit is that the GMM can model a complex, globally nonlinear manifold well with a collection of *locally linear* models that exploit the tractability of operations in the Gaussian domain. Previous researchers used the GMMs for the voice conversion problem because of the intuitive notion that the individual component densities models the underlying set of acoustic classes [41]. It is assumed that the acoustic space corresponding to a speaker's voice can be characterized by a set of acoustic classes representing some broad phonetic events, such as vowels, nasals, or fricatives. The GMM has been used successfully for both speaker identification [41] and voice conversion [37, 2].

Contrary to classification schemes with "hard" class boundaries, data points have varying degrees of "membership" to all local models; this is referred to as "soft" partitioning. The conditional probability of a GMM class j given $\mathbf{z}$ is derived by the

application of Bayes' rule

$$p(j/\mathbf{z}) = \frac{p(\mathbf{z}/j)P(j)}{p(\mathbf{z})} \tag{3.7}$$
$$= \frac{p(\mathbf{z}/j)P(j)}{\sum_{j=1}^{M} P(\mathbf{z}/j)P(j)} \tag{3.8}$$

The GMM parameters $\lambda = \{\{P(j), \boldsymbol{\mu}_j, \Sigma_j\},\ j = 1, ..., M\}$ are estimated by application of an Expectation Maximization (EM) algorithm [42, 41], an iterative method for computing the maximum likelihood parameter estimates. For a sequence of N training vectors $Z = \{\mathbf{z_1}, ...\mathbf{z_N}\}$, the GMM likelihood can be written as

$$p(Z/\lambda) = \prod_{t=1}^{N} p(z_t/\lambda) \tag{3.9}$$

Initially, the mixing weights $P(j)$, $\boldsymbol{\mu}$ are initialized using the K-means clustering algorithm, and covariances Σ equal to the identity matrix. Based on this initial model the next iteration is carried out.

On each EM iteration, the following reestimation formulas are used which guarantee a monotonic increase in the model's likelihood value:

Mixture weights:

$$P(j) = \frac{1}{N} \sum_{n=1}^{N} P(j|\mathbf{z}_n, \lambda) \tag{3.10}$$

Means:

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_{n=1}^{N} P(j|\mathbf{z}_n, \lambda)\mathbf{z}_n}{\sum_{n=1}^{N} P(j|\mathbf{z}_n, \lambda)} \tag{3.11}$$

Covariances:

$$\hat{\Sigma}_j = \frac{\sum_{n=1}^{N} P(j|\mathbf{z}_n, \lambda)|\mathbf{z}_n - \hat{\mu}_j||\mathbf{z}_n - \hat{\mu}_j|^T}{\sum_{n=1}^{N} P(j|\mathbf{z}_n, \lambda)} \tag{3.12}$$

The *a posteriori* probability for acoustic class i is given by Equation 3.8. The new model then becomes the initial model for the next iteration and the process is repeated until some convergence threshold is reached. It may be observed that Equation 3.12 is the limiting operation computationally because it involves the complete reestimation of the j^{th} sample covariance matrix weighted by the j^{th} posterior component probability. Thus, the EM algorithm's complexity with fully populated covariance matrices is $O(NMd^2)$.

3.2.2 Spectral Conversion with a GMM

Having estimated the parameters of the GMM, we can now estimate the target speakers feature vector $\mathbf{y}$ from the source speakers feature vector $\mathbf{x}$. The joint covariance matrix Σ_j for the j^{th} Gaussian component is partitioned as follows.

$$\Sigma_j = \begin{bmatrix} \Sigma_j^{xx} & \Sigma_j^{xy} \\ \Sigma_j^{yx} & \Sigma_j^{yy} \end{bmatrix} \tag{3.13}$$

where

Σ_j^{xx} is the $\frac{d}{2} \times \frac{d}{2}$ auto-covariance of the source vector $\mathbf{x}$.

$$E[[\mathrm{x} - \mu_j^x][\mathrm{x} - \mu_j^x]^T] \tag{3.14}$$

Σ_j^{xy} is the $\frac{d}{2} \times \frac{d}{2}$ cross-covariance of the source vector $\mathbf{x}$ with the target vector $\mathbf{y}$.

$$E[[\mathrm{x} - \mu_j^x][\mathrm{y} - \mu_j^y]^T] \tag{3.15}$$

Σ_j^{yx} is the $\frac{d}{2} \times \frac{d}{2}$ cross-covariance of the target vector $\mathbf{y}$ with the source vector $\mathbf{x}$. It is the transpose of Equation 3.15. Σ_j^{yy} is the $\frac{d}{2} \times \frac{d}{2}$ auto-covariance of the target vector

y.

$$E[[\mathrm{y} - \mu_j^y][\mathrm{y} - \mu_j^y]^T] \tag{3.16}$$

The expected value of a feature vector $\mathbf{y}$ given feature vector x for one component single gaussian is the regression

$$\begin{aligned} E[\mathrm{y}|\mathrm{x}] &= \int \mathrm{y}p(\mathrm{y}|\mathrm{x})dy & (3.17) \\ &= \mu_j^y + \Sigma_j^{yx}(\Sigma_j^{xx})^{-1}(\mathrm{x} - \mu_j^x) & (3.18) \end{aligned}$$

Extending to the mixture case, the expectation is

$$F(\mathrm{x}) = E[\mathrm{y}|\mathrm{x}] = \sum_{j=1}^{M} P(j|\mathrm{x})[\mu_j^y + \Sigma_j^{yx}(\Sigma_j^{xx})^{-1}(\mathrm{x} - \mu_j^x)] \tag{3.19}$$

with

$$\mu_{\mathrm{j}} = \begin{bmatrix} \mu_j^x \\ \\ \mu_j^y \end{bmatrix} \tag{3.20}$$

as shown in Kambhatla's work on Gaussian mixture models for statistical data processing [43]. The Equation 3.19 is referred to as the conversion function [37].

3.3 Transformation

In the transformation mode, the system analyzes a test speech file of the source speaker and transforms the extracted features x to $\hat{\mathrm{y}}$, an estimate of the target speaker's LSF parameters.

For each frame we calculate the transformed spectral envelope by converting the estimated LSF parameters back to LPC filter coefficients which in turn are used for

synthesis. In addition, pitch scaling is employed to match the pitch of the source speaker to that of the target speaker.

3.4 Voice Conversion using Factor Analysis

Although the GMM has become quite popular in recent times for modelling complex probability densities, it has a few shortcomings. The use of a GMM with full co-variance matrices leads to a huge number of parameters for a high-dimensional input space and presents the risk of over-fitting. Covariance matrices can at the most be constrained to be diagonal. The latter constraint leads to a model in which the axes of the Gaussians are aligned with the data axes and which does not capture correlation amongst the variables. Thus, each of these parameterizations has its disadvantages. With full covariance.matrices, each EM step requires $O(NMd^2)$ operations, where N is the number of vectors in data space, M is the number of components in the mixture model and d is the dimension of the data space. Diagonal covariance matrices limit the computational complexity to $O(NMd)$ and restrict the amount of data needed for reliable estimation.

A compromise between these extremes can be found in the recently introduced mixture of latent variable models [48, 49] which form a mixture of constrained Gaussians. The advantage of using mixtures of latent variable models is that one can avoid the constraint of aligned axes (thus capturing correlations) without needing a full covariance matrix. This can be done by using the freedom we have in choosing the dimension

of the so-called latent space: the covariance matrices of the Gaussians are specified and controlled through a mapping from this latent space to the data space. This idea is illustrated in Figure 3.3.

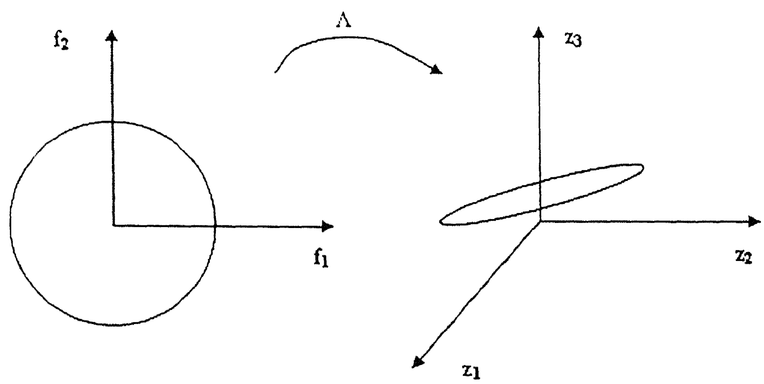

Figure 3.3: A generative model from a latent space of dimension 2 to a data space of dimension 3

A latent variable model relates a d-dimensional observed data vector $\mathbf{z}$ to a q-dimensional ($q < d$) latent vector $\mathbf{f}$ by defining a noise model and a prior on the distribution of the latent variables. Recently, there has been a great deal of research on the topic of local dimensionality reduction, resulting in several variants of the basic concept with successful applications to character recognition [50]. The algorithm used by these authors for dimensionality reduction is Principal Component Analysis (PCA). PCA, unlike maximum likelihood factor analysis (FA), does not define a proper density model for the data [52]. Furthermore, PCA is not robust to independent noise in the features of the data. The Mixture of Factor Analyzers (MFA) first proposed by Zoubin

Ghahramani [49] and then by McLachlan [53] can be used to solve the inflexibility of GMMs and achieve local dimensionality reduction in each cluster. Comparison studies by Moerland [51], proved that Mixture of Factor Analyzers outperform Mixtures of Principal Component Analyzers [48].

3.4.1 Factor Analysis

In maximum likelihood factor analysis, a d-dimensional real-valued data vector $\mathbf{z}$ is modelled using a q-dimensional vector of real-valued factors, $\mathbf{f}$, where q is generally much smaller that d [54]. The generative model is given by:

$$\mathbf{z} = \Lambda\mathbf{f} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{3.21}$$

where Λ is known as the *factor loading matrix*. The factors $\mathbf{f}$ are assumed to be $N(0, I)$ distributed (zero-mean independent normals, with unit variance). The d-dimensional random variable $\boldsymbol{\epsilon}$ is distributed $N(0, \psi)$, where ψ is a diagonal matrix. The diagonality of ψ is one of the key assumptions of factor analysis: the observed variables are independent given the factors. The term $\boldsymbol{\mu}$ represents the non zero mean which the data can assume. Under these assumptions, the observations $\mathbf{z}$ are Gaussian with mean

$$\begin{aligned} E[\mathbf{z}] &= E[\Lambda\mathbf{f} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] && (3.22) \\ &= \Lambda E[\mathbf{f}] + E[\boldsymbol{\mu}] + E[\boldsymbol{\epsilon}] && (3.23) \\ &= \boldsymbol{\mu} && (3.24) \end{aligned}$$

and model covariance C.

$$\begin{aligned} C &= E[[\mathbf{z}-\boldsymbol{\mu}][\mathbf{z}-\boldsymbol{\mu}]^T] && (3.25)\\ &= E[[\Lambda\mathbf{f}+\epsilon][\Lambda\mathbf{f}+\epsilon]^T] && (3.26)\\ &= \Lambda E[\mathbf{f}\mathbf{f}^T]\Lambda^T + E[\epsilon\mathbf{f}^T]\Lambda^T + \Lambda E[\mathbf{f}\epsilon^T] + E[\epsilon\epsilon^T] && (3.27)\\ &= \Lambda\Lambda^T + \psi && (3.28) \end{aligned}$$

Given Λ and ψ, the expected value of the factors can be computed through the linear projection

$$E[\mathbf{f}|\mathbf{z}] = \beta(\mathbf{z}-\boldsymbol{\mu}) \tag{3.29}$$

where $\beta \equiv \Lambda'(\psi + \Lambda\Lambda')^{-1}$, a fact that results from the joint normality of data and factors.

$$P\left(\begin{bmatrix}\mathbf{z}\\ \mathbf{f}\end{bmatrix}\right) = N\left(\begin{bmatrix}\boldsymbol{\mu}_z\\ 0\end{bmatrix}, \begin{bmatrix}\Lambda\Lambda'+\psi & \Lambda\\ \Lambda' & I\end{bmatrix}\right) \tag{3.30}$$

where I is an identity matrix. Furthermore, it is possible to compute the second moment of the factors [49],

$$\begin{aligned} E[\mathbf{f}\mathbf{f}'|\mathbf{z}] &= Var(\mathbf{f}|\mathbf{z}) + E[\mathbf{f}|\mathbf{z}]E[\mathbf{f}|\mathbf{z}]' && (3.31)\\ &= I - \beta\Lambda + \beta(\mathbf{z}-\boldsymbol{\mu})(\mathbf{z}-\boldsymbol{\mu})'\beta' && (3.32) \end{aligned}$$

With this model, the common factors $\mathbf{f}$ account for the statistical dependencies between the individual variables of $\mathbf{z}$, and the specific factors ϵ explain small disturbances in each individual random variable of $\mathbf{z}$. In our model for voice conversion, the

common factors f capture the correlations between LSFs, and the specific factors capture the sensor noise about each individual LSF. Capturing these correlations allows us to eliminate the redundancies in the LSFs.

3.4.2 Mixture of Factor Analyzers (MFA)

In this work, we follow the EM algorithm proposed by Ghahramani [49]. We give a brief overview of the mixture of factor analyzers in this section.

We assume we have a mixture of M factor analyzers indexed by $\omega_j, j = 1, ...M$. The generative model now obeys the following mixture distribution:

$$P(\mathbf{z}) = \sum_{j=1}^{M} \int P(\mathbf{z}|\mathbf{f}, \omega_j) P(\mathbf{f}|\omega_j) P(\omega_j) d\mathbf{f} \tag{3.33}$$

As in regular factor analysis, the factors are assumed to be $N(0, I)$ distributed, therefore

$$P(\mathbf{f}|\omega_j) = P(\mathbf{f}) = N(0, I) \tag{3.34}$$

The idea behind the mixture of factor analyzers is illustrated in Figure 3.4. Each factor analyzer in the mixture has a different mean $\boldsymbol{\mu}_j$ and covariance Σ_j. Therefore,

$$P(\mathbf{z}|\mathbf{f}, \omega_j) = N(\boldsymbol{\mu}_j + \Lambda_j \mathbf{f}, \psi) \tag{3.35}$$

As discussed in §3.2.1, data $\mathbf{z}$ is given by the Expression 3.3.

The parameters of this model are $\{(\boldsymbol{\mu}_j, \Lambda_j)_{j=1}^{M}, \boldsymbol{\pi}, \psi\}$; the vector $\boldsymbol{\pi}$ parameterizes the adaptable mixing proportions, $\pi_j = P(\omega_j)$. The latent variables in this model are the factors $\mathbf{f}$ and the mixture indicator variable ω, where $w_j = 1$ when the data

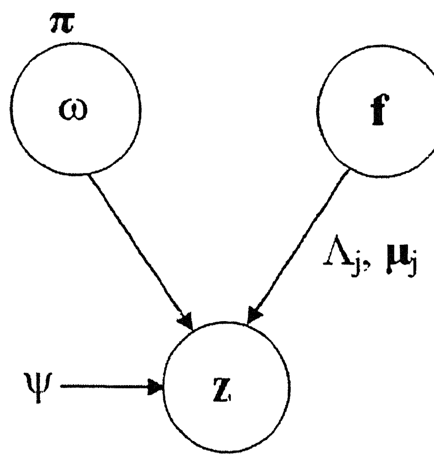

Figure 3.4: The mixture of factor analysis generative model

point was generated by ω_j. For the E-step of the EM algorithm, one needs to compute expectations of all the interactions of the hidden variables that appear in the log likelihood. The following statements hold true and can be verified,

$$E[w_j\mathbf{f}|\mathbf{z}_i] = E[w_j|\mathbf{z}_i]\; E[\mathbf{f}|w_j, \mathbf{z}_i] \tag{3.36}$$

$$E[w_j\mathbf{f}|\mathbf{z}_i] = E[w_j|\mathbf{z}_i]\; E[\mathbf{f}|w_j, \mathbf{z}_i] \tag{3.37}$$

Defining

$$h_{ij} = E[w_j|\mathbf{z}_i] = \pi_j N(\mathbf{z}_i - \boldsymbol{\mu}_j, \Lambda\Lambda' + \psi) \tag{3.38}$$

and using Equations 3.29 and 3.36 we obtain

$$E[w_j\mathbf{f}|\mathbf{z}_i] = h_{ij}\beta_j(\mathbf{z}_i - \boldsymbol{\mu}_j) \tag{3.39}$$

where $\beta_j = \Lambda_j'(\psi + \Lambda_j\Lambda_j')^{-1}$. Similarly using Equations 3.32 and 3.37, the following expression is obtained

$$E[w_j\mathbf{f}\mathbf{f}'|\mathbf{z}_i] = h_{ij}(I - \beta_j\Lambda_j + \beta_j(\mathbf{z}_i - \boldsymbol{\mu}_j)(\mathbf{z}_i - \boldsymbol{\mu}_j)'\beta_j') \tag{3.40}$$

The expected log likelihood for mixture of factor analyzers is

$$Q = E[log \prod_i \prod_j \{(2\pi)^{d/2}|\psi|^{-1/2} exp\{-\frac{1}{2}[\mathbf{z}_i - \boldsymbol{\mu}_j - \Lambda_j \mathbf{f}]'\psi^{-1}[\mathbf{z}_i - \boldsymbol{\mu}_j - \Lambda_j \mathbf{f}]\}\}^{w_j}] \quad (3.41)$$

The EM algorithm for mixtures of factor analyzers is briefly discussed:

E-Step: Compute h_{ij}, $E[\mathbf{f}|\mathbf{z}_i, \omega_j]$ and $E[\mathbf{ff}'|\mathbf{z}_i, \omega_j]$ for all data points i and mixture components j.

M-Step: The reestimation formulae for $\pi_j, \Lambda_j, \boldsymbol{\mu}_j$ and ψ are obtained by maximizing Equation 3.41. These are listed below:

$$\pi_j = \frac{1}{N}\sum_{i=1}^{N} h_{ij} \quad (3.42)$$

$$[\Lambda_j \quad \mu_j] = \left(\sum_i h_{ij}\mathbf{z}_i E[\tilde{\mathbf{f}}|\mathbf{z}_i, \omega_j]'\right)\left(\sum_l h_{lj} E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{z}_l, \omega_j]'\right)^{-1} \quad (3.43)$$

where

$$E[\tilde{\mathbf{f}}|\mathbf{z}_i, \omega_j] = \begin{bmatrix} E[\mathbf{f}|\mathbf{z}_i, \omega_j] \\ 1 \end{bmatrix} \quad (3.44)$$

and

$$E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{z}_l, \omega_j] = \begin{bmatrix} E[\mathbf{ff}'|\mathbf{z}_l, \omega_j] & E[\mathbf{f}|\mathbf{z}_l, \omega_j] \\ E[\mathbf{f}|\mathbf{z}_l, \omega_j]' & 1 \end{bmatrix} \quad (3.45)$$

The reestimation formula for ψ is given by the expression

$$\psi = \frac{1}{n} diag \left\{\sum_{ij} h_{ij}\left(\mathbf{z}_i - \tilde{\Lambda}_j E[\tilde{\mathbf{f}}|\mathbf{z}_i, \omega_j]\right)\mathbf{z}'_i\right\} \quad (3.46)$$

The mixture of factor analyzers is, in essence, a reduced dimensionality mixture of Gaussians. Each factor analyzer fits a Gaussian to a portion of the data, weighted by the posterior probabilities, h_{ij}. Since the covariance matrix for each Gaussian is

specified through the lower dimensional factor loading matrices, the model has $Mqd+d$, rather than $Md(d+1)/2$, parameters dedicated to modelling covariance structure.

3.4.3 Use of Unaligned Data to Improve Performance

The method discussed is the previous sections, require the source data in addition to its corresponding aligned target data. There are many applications of voice conversion that, in the training step, more data from the source speaker is available than of the target. For example, if we are going to personalize the output of a Text to Speech Synthesizer system sentences can be generated of the source speaker. However the target data is fixed. This section discusses a method to incorporate this extra unaligned data in the training phase.

Previous studies [56] have shown that, including unlabelled data in classification problems leads to an increase in the performance of the classification. We apply this idea to the voice conversion problem.

The expected likelihood function for the mixture of factor analyzers in Equation 3.41 is modified to include unaligned data $\mathbf{x}_k$.

$$Q = E[log \prod_{i,j} P(\mathbf{x}_i, \mathbf{y}_i | \mathbf{f}, \omega_j) \prod_{k,j} P(\mathbf{x}_k | \mathbf{f}, \omega_j)] \tag{3.47}$$

Substituting for $P(\mathbf{x}_i, \mathbf{y}_i|\mathbf{f}, \omega_j)$ and $P(\mathbf{x}_k|\mathbf{f}, \omega_j)$ from Equation 3.35 we get,

$$\begin{aligned} Q &= E\Bigg[log \prod_{i,j} \left\{(2\pi)^{d/2}|\psi|^{-1/2} exp\{\frac{-1}{2}[\mathbf{z}_i - \boldsymbol{\mu}_j - \Lambda_j \mathbf{f}]'\psi^{-1}[\mathbf{z}_i - \boldsymbol{\mu}_j - \Lambda_j \mathbf{f}]\}\right\}^{w_j} \\ &\prod_{k,j} \left\{(2\pi)^{d/4}|\psi_x|^{-1/2} exp\{\frac{-1}{2}[\mathbf{x}_k - \boldsymbol{\mu}_{jx} - \Lambda_j \mathbf{f}]'\psi_x^{-1}[\mathbf{x}_k - \boldsymbol{\mu}_{jx} - \Lambda_j \mathbf{f}]\}\right\}^{w_j}\Bigg] \end{aligned} \tag{3.48}$$

To initialize the model, first the parameters of the MFA model are estimated from aligned data as in §3.4.2. The following assumption is made for simplicity. The matrix ψ, i.e. the covariance structure of ϵ is not affected by addition of extra unaligned data. An EM algorithm can now be derived from Expression 3.48 to recalculate the means, factor loading matrices and the weights without modifying ψ. The problem now comprises of obtaining a better estimate of the mean vector $\boldsymbol{\mu}_j$, the factor loading matrices Λ_j and the mixing weights $P(\omega_j)$.

To jointly estimate $\boldsymbol{\mu}_j$ and the factor loading matrices Λ_j let

$$\tilde{\mathbf{f}} = \begin{bmatrix} \mathbf{f} \\ 1 \end{bmatrix} \tag{3.49}$$

$$\tilde{\Lambda}_j = \begin{bmatrix} \Lambda_j & \boldsymbol{\mu}_j \end{bmatrix} = \begin{bmatrix} \tilde{\Lambda}_{jx} \\ \tilde{\Lambda}_{jy} \end{bmatrix} = \begin{bmatrix} \Lambda_{jx} & \boldsymbol{\mu}_{jx} \\ \Lambda_{jy} & \boldsymbol{\mu}_{jy}. \end{bmatrix} \tag{3.50}$$

Rewriting, Equation 3.48 with the above substitutions and simplifying, we get

$$\begin{aligned} Q = E\Bigg[log \prod_{i,j} \Big\{ (2\pi)^{d/2} |\psi|^{-1/2} exp\{\frac{-1}{2}[\mathbf{z}_i - \tilde{\Lambda}_j \tilde{\mathbf{f}}]' \psi^{-1} [\mathbf{z}_i - \tilde{\Lambda}_j \tilde{\mathbf{f}}]\} \Big\}^{w_j} \\ \prod_{k,j} \Big\{ (2\pi)^{d/4} |\psi_x|^{-1/2} exp\{\frac{-1}{2}[\mathbf{x}_k - \tilde{\Lambda}_j \tilde{\mathbf{f}}]' \psi_x^{-1} [\mathbf{x}_k - \tilde{\Lambda}_j \tilde{\mathbf{f}}]\} \Big\}^{w_j} \Bigg] \end{aligned} \tag{3.51}$$

$$\begin{aligned} = {} & c - \frac{N_1}{2} log|\psi| - \frac{N_2}{2} log|\psi_x| \\ & - \sum_{i,j} \frac{1}{2} h_{ij} \mathbf{z}_i' \psi^{-1} \mathbf{z}_i - h_{ij} \mathbf{z}_i' \psi^{-1} \tilde{\Lambda}_j E[\tilde{\mathbf{f}} | \mathbf{z}_i, \omega_j] + \frac{1}{2} h_{ij} tr[\tilde{\Lambda}_j' \psi^{-1} \tilde{\Lambda}_j E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}' | \mathbf{z}_i, \omega_j] \\ & - \sum_{k,j} \frac{1}{2} h_{kj} \mathbf{x}_k' \psi^{-1} \mathbf{x}_k - h_{kj} \mathbf{x}_k' \psi^{-1} \tilde{\Lambda}_j E[\tilde{\mathbf{f}} | \mathbf{x}_k, \omega_j] + \frac{1}{2} h_{kj} tr[\tilde{\Lambda}_j' \psi^{-1} \tilde{\Lambda}_j E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}' | \mathbf{x}_k, \omega_j] \end{aligned} \tag{3.52}$$

where c is a constant, $h_{ij} = P(\omega_j | \mathbf{z}_i)$ (from Equation 3.39) and N_1 and N_2 are the

number of vectors in the aligned data $\mathbf{z}_i$ and the unaligned source data $\mathbf{x}_k$ respectively.

To estimate $\tilde{\Lambda}_{jx}$ and $\tilde{\Lambda}_{jy}$, we set $\frac{\partial Q}{\partial \tilde{\Lambda}_{jx}} = 0$ and $\frac{\partial Q}{\partial \tilde{\Lambda}_{jy}} = 0$

$$\begin{aligned}
\frac{\partial Q}{\partial \tilde{\Lambda}_{jx}} &= -\sum_i -h_{ij}\psi_x^{-1}\mathbf{x}_i E[\tilde{\mathbf{f}}|\mathbf{z}_i,\omega_j]' + h_{ij}\psi_x^{-1}\tilde{\Lambda}_{jx}E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{z}_i,\omega_j] && (3.53)\\
&\quad -\sum_k -h_{kj}\psi_x^{-1}\mathbf{x}_k E[\tilde{\mathbf{f}}|\mathbf{x}_k,\omega_j]' + h_{kj}\psi_x^{-1}\tilde{\Lambda}_{jx}E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{x}_k,\omega_j]
\end{aligned}$$

Solving for $\tilde{\Lambda}_{jx}$, we obtain

$$\begin{aligned}
\tilde{\Lambda}_{jx} &= \left[\sum_i h_{ij}\mathbf{x}_i E[\tilde{\mathbf{f}}|\mathbf{z}_i,\omega_j]' + \sum_k h_{kj}\mathbf{x}_k E[\tilde{\mathbf{f}}|\mathbf{x}_k,\omega_j]'\right] && (3.54)\\
&\quad \times \left[\sum_i h_{ij}E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{z}_i,\omega_j] + \sum_k h_{kj}E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{x}_k,\omega_j]\right]^{-1}
\end{aligned}$$

Similarly,

$$\frac{\partial Q}{\partial \tilde{\Lambda}_{jy}} = -\sum_i -h_{ij}\psi_y^{-1}\mathbf{y}_i E[\tilde{\mathbf{f}}|\mathbf{z}_i,\omega_j]' + h_{ij}\psi_y^{-1}\tilde{\Lambda}_{jy}E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{z}_i,\omega_j] = 0 \quad (3.55)$$

Therefore,

$$\tilde{\Lambda}_{jy} = \left[\sum_i h_{ij}\mathbf{y}_i E[\tilde{\mathbf{f}}|\mathbf{z}_i,\omega_j]'\right]\left[\sum_l h_{lj}E[\tilde{\mathbf{f}}\tilde{\mathbf{f}}'|\mathbf{z}_l,\omega_j]\right]^{-1} \quad (3.56)$$

To reestimate the mixing weights, using the definition for $P(\omega_j)$ and the empirical distribution of data as an estimate of $P(\mathbf{x})$

$$\begin{aligned}
P(\omega_j) &= \int P(\omega_j|\mathbf{x})P(\mathbf{x})d\mathbf{x} && (3.57)\\
&= \frac{1}{N_1+N_2}\left(\sum_{i=1}^{N_1} h_{ij} + \sum_{k=1}^{N_2} h_{kj}\right) && (3.58)
\end{aligned}$$

3.4.4 Spectral Conversion with MFAs

In order to convert the spectrum in the Factor Analysis case, we calculate the expectation of the target vector $\mathbf{y}$ given the source vector $\mathbf{x}$ for the single Factor Analysis model

and then extend the result to a Mixture of Factor Analyzers. To find this expectation, we partition $\mathbf{z}$ so that its joint multivariate density is

$$P\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = N\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \Sigma\right) \tag{3.59}$$

where Σ is the covariance matrix given by

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \tag{3.60}$$

From Equation 3.28 the terms Σ_{xx} and Σ_{yy} are given by

$$\Sigma_{xx} = \Lambda_x \Lambda_x^T + \psi_x \tag{3.61}$$

$$\Sigma_{yy} = \Lambda_y \Lambda_y^T + \psi_y \tag{3.62}$$

Equation 3.15 leads to the following expressions for Σ_{xy} and Σyx

$$\Sigma_{xy} = \Lambda_x \Lambda_y^T \tag{3.63}$$

$$\Sigma_{yx} = \Lambda_y \Lambda_x^T \tag{3.64}$$

Using Equation 3.18, the conditional expectation of a joint Gaussian, we find that

$$E[\mathbf{y}|\mathbf{x}] = \Lambda_y \Lambda_x^T (\Lambda_x \Lambda_x^T + \psi_x)^{-1} (\mathbf{x} - \boldsymbol{\mu}_x) + \boldsymbol{\mu}_y \tag{3.65}$$

Extending the result to the mixture case, the statistical mapping becomes

$$E[\mathbf{y}|\mathbf{x}] = \sum_{j=1}^{M} P(\omega_j|\mathbf{x}) [\Lambda_y \Lambda_x^T (\Lambda_x \Lambda_x^T + \psi_x)^{-1} (\mathbf{x} - \boldsymbol{\mu}_{jx}) + \boldsymbol{\mu}_{jy}] \tag{3.66}$$

In fitting a mixture of factor analyzers the modeler thus has two free parameters to decide: the number of factor analyzers to use (M), and the number of factors in each

analyzer (q). These parameters can be chosen empirically based on the quality of the synthesized speech.

Chapter 4

Evaluation

4.1 Speech Corpus

For our research, we specifically chose the Arctic Speech Corpus [11] developed recently by Carnegie Mellon University(CMU) which is a high quality speech corpus with a free software license. It was specifically created for the purpose of speech synthesis.

Since this thesis only deals with spectral conversion, the Arctic corpus suits our purpose. The CMU Arctic Speech Corpus is a set of single speaker databases that have been carefully recorded under studio conditions. The databases consist of around 1150 phonetically rich sentences carefully selected from out-of-copyright texts from Project Gutenberg [27]. The corpus consists of four primary sets of recordings (3 male, 1 female). The corpus consists of *16 bit* speech 'wav' files with a sampling rate of *16 khz.*

4.2 Objective Evaluation

In evaluating our system, we use the objective measure given by Kain [5]. This performance index is a ratio of two measures. The first measure, the *transspeaker distance*, is the spectral distance between the converted speech and the target speech determining how close the converted speech is to the target speakers. The second, the *interspeaker distance*, measures the spectral distance between the source and target speaker. To present the performance index, let us again consider the vector of source speech for the n^{th} frame as $\mathbf{x}_n$ and the target speakers n^{th} vector as $\mathbf{y}_n$. We compute the performance index P with the following equation

$$P = 1 - \frac{\sum_{n=1}^{N} D(\mathbf{y}_n, \hat{\mathbf{y}}_n)}{\sum_{n=1}^{N} D(\mathbf{y}_n, \mathbf{x}_n)} \tag{4.1}$$

where $\hat{\mathbf{y}}_n$ is the converted target vector. Since our feature for representing speech is the LSF, the distance measure D between any two LSF vectors $\mathbf{a}$ and $\mathbf{b}$ with dimension p is Euclidean.

$$\left\{ \sum_{k=1}^{p} (a_k - b_k)^2 \right\}^{\frac{1}{2}} \tag{4.2}$$

We interpret a performance index close to 1 as a good conversion while one close to 0 is not performing well. Equation 4.1 is close to 1 when the source and target speaker have different spectral properties and when the converted speech is close to the target speakers speech. Thus, according to the objective measure, it is difficult for the system to perform well if the source and target speakers spectral properties are already similar.

Figure 4.1 shows the graph of performance index versus the training time for a

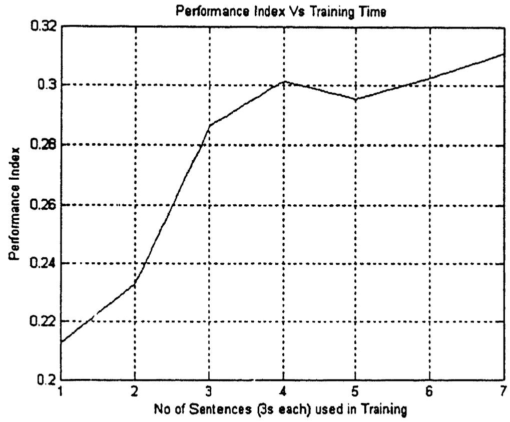

Figure 4.1: Performance Index versus Training Time for conversion using mixture of factor analyzers

voice conversion system using mixture of factor analyzers with 8 components. As the training time increases, the performance of each system exponentially increases until it plateaus around 15 seconds of training data. We conclude that 15 seconds of training data (about 5 sentences of 3s each) is enough for the system to give a reasonable average performance between 0.25 and 0.30.

Figure 4.2 shows the variation of performance index for different number of components chosen in the mixture model for three cases viz. mixture of factor analyzers, mixture of factor analyzers with unaligned data and for full covariance GMM. The dimension of the latent variable **f** was chosen as *8* in this experiment. It can be seen that GMM performs much better than MFA for components greater than *12*. When unaligned data is used in the EM training, this system outperforms the previous two cases. The performance drops when the number of components is increased because of overfitting.

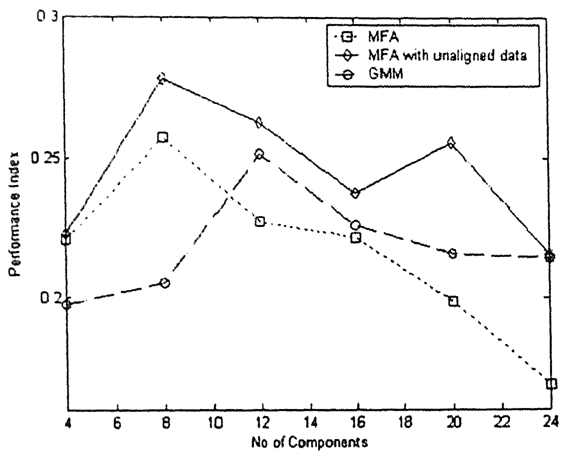

Figure 4.2: Performance Index versus number of components in the mixture model

Figure 4.4 displays an example of envelope conversion for the case using a mixture of factor analyzers, the case when unaligned data is included and is bench marked against the full covariance GMM method. It can be seen that there is a shift in the formants of the converted envelopes, and the closest fit to the target envelope is achieved when unaligned data is used in the training phase.

4.3 Subjective Tests

Subjective listening tests were conducted with seven listeners to assess the recognizability and quality of the converted speech for the system using a mixture of factor analyzers with and without unaligned data along with the system using GMM. Before starting the test, we trained each listener on each of the four speakers' distinct voice qualities by playing several of their speech files. After this initial training, we quizzed each listener until they identified each speaker correctly for ten consecutive

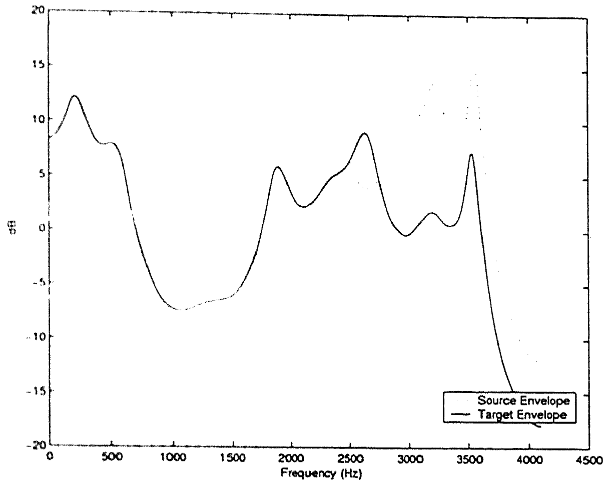

Figure 4.3: Source and Target Envelopes

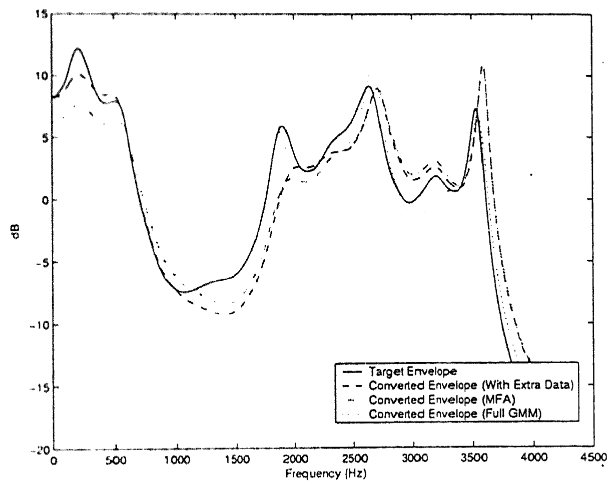

Figure 4.4: Comparison of converted envelopes with actual target envelope

trials. Three kinds of listening tests have been conducted; ABX test, speaker discrimination test and system quality comparison test. After quizzing, we began the test, playing the same sentence "God your letter came just in time" for different cases. Between each utterance, we gave the listener 10 seconds to write down his answer; these time intervals made the entire test last about 50 minutes. We played the same sentence so that listeners could focus intently on the quality of each recording.

ABX Test:

To evaluate the accuracy of the conversion, a set of trials were presented to the listeners using the ABX method. X was either the converted speech by using GMM with 12 components or the converted speech using MFA with or without unaligned data using 8 components. A and B were either the target or the source speaker. Speakers A and B uttered the same sentence which in general was different from the sentence uttered by X. Subjects were asked to select either A or B as being most similar to X. Figure 4.5

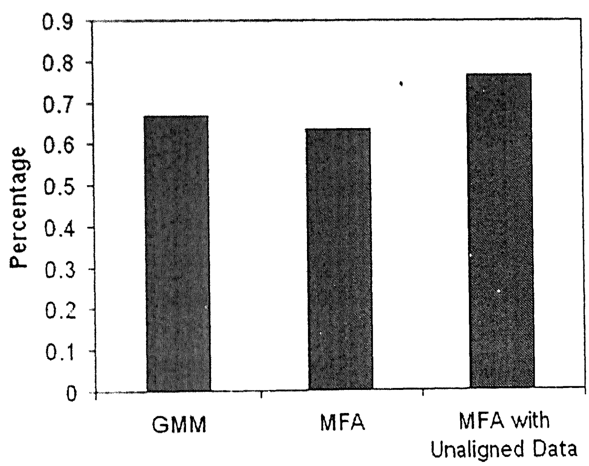

Figure 4.5: Results from the ABX Test

summarizes the results from this test giving the percentage of correct answers. A correct answer means that the converted/ modified speaker was recognized as the target speaker. Conversion obtained using only MFA scored 63% and was marginally below that obtained by GMM. However, when unaligned data was used in training 76% of the time, the correct answers were received.

System Quality Comparison Test:

To assess the speech quality of the various voice conversion systems in terms of intelligibility and naturalness, we compared them against each other. The listeners were asked to give their preference scores on a scale of 1-5, 1 being the lowest score and 5 being the highest, for various samples of the converted speech. The listeners preferences are

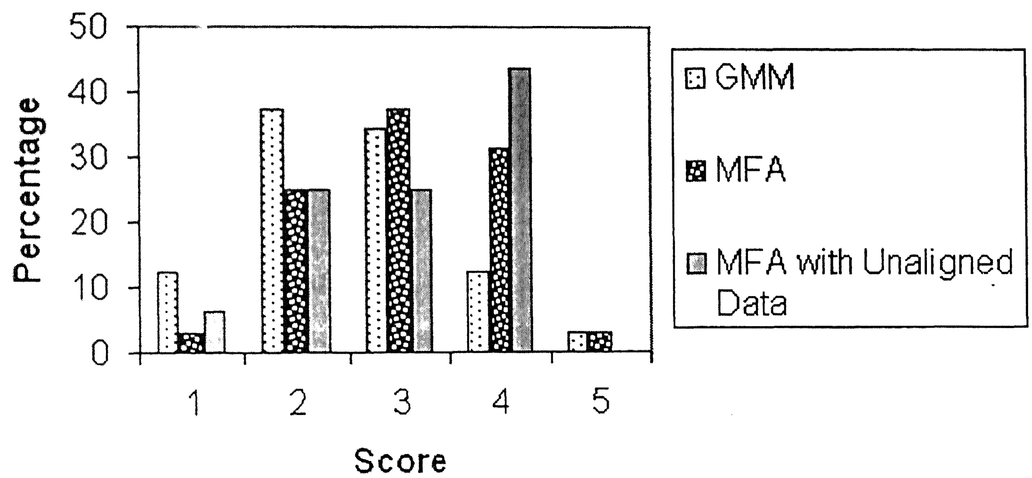

Figure 4.6: System Quality Comparison Test

shown in Figure 4.6. The overall quality of the converted signals was considered as quite natural, although some of the listeners reported a muffling effect in some cases.

Listeners scored a higher preference to the converted speech using MFA with unaligned data.

Speaker Discrimination Test:

In this test, the listeners were asked to recognize the speakers. The converted speech samples obtained from the three methods were played randomly. The recognition accuracy was about 72% for GMM and the MFA using unaligned data and was marginally lesser (68.5%) for the conversion obtained using only MFA.

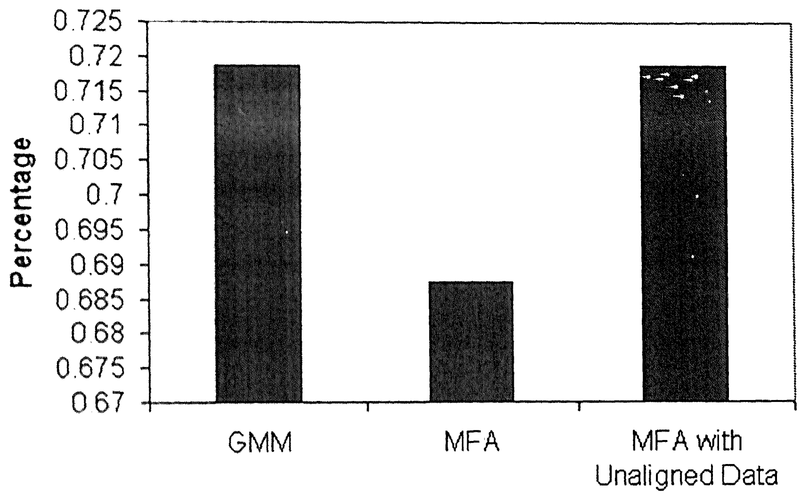

Figure 4.7: System Discrimination Test

Listeners had difficulty in distinguishing between two male speakers in the database. Listeners were also presented samples of converted speech obtained using different number of dimensions in the factor analyzers. None of the subjects could differentiate between the samples when the differences in the dimensions were small. When modelling the probability density with a mixture of factor analyzers, we are thus able to

present a finer resolution of options to the end user of the voice conversion system. The end user can select with a greater degree of freedom how well the system should perform depending on the application.

Chapter 5

Conclusion

5.1 Summary

In summary, we have applied the mixture of factor analyzers model to voice conversion. The MFA model has $Mqd + q$, rather than $Md(d+1)/2$ parameters dedicated to modelling the covariance structure. The time complexity for training is reduced from $O(MNd^2)$ to $O(MNdq)$. This reduction in complexity comes with the penalty of a slight decrease in the objective quality of conversion. It has also been shown that a combined learning with aligned source - target speaker data and unaligned source data increases the conversion performance. Subjective tests showed that, small changes in the dimension of the factor analyzers did not affect the perception of the speech. Therefore user of the system can select an appropriate value of the dimension of the factor analyzer to suit his performance needs of the application. For voice conversion training and execution, the mixture of factor analyzers model provides a flexible range of tradeoffs to select from.

5.2 Future Work

The system can be improved upon in several ways by incorporating the recent advances of variational Bayesian modelling, independent component analysis and on line EM algorithms.

By incorporating variational Bayesian techniques with MFA as described in [57], the system can can automatically determine the optimal number of components and the local dimensionality of each component (i.e. the number of factors in each factor analyzer). This method falls in the automatic relevance determination framework of variational Bayesian techniques. Having this ability implies that we can describe the nonlinear probability density with an appropriate number of components for the mixture. Incorporating this technique increases the complexity slightly but solves the difficult problem of estimating the correct model order and dimensionality of each component and alleviates the end user of these responsibilities.

The EM algorithm used here is a batch algorithm in which the whole training data is scanned at every iteration to improve its estimation. Including new test data in the model requires a complete reestimation of the model parameters. In addition, memory space required by the algorithm should be constant with respect to number of data points processed so far. Online EM algorithms can be thus used overcome these problems and the model can be updated online based on new training data.

Using a Gaussian for each component may not always hold true in natural clustering problems. One way to solve this problem is to use a mixture of independent component

analyzers as described in [55] where each component's distribution is non-Gaussian. With this approach, it may be possible to describe the density of each component in the mixture more accurately.

References

[1] E. Moulines and Y. Sagiska, Eds., "Voice Conversion: State of the Art and Perspectives", Special Issue of Speech Communication, vol. 16, Feb. 1995.

[2] A. Kain and M. Macon, "Spectral voice conversion for text-to-speech synthesis", In Proceedings of ICASSP '98, vol. 1, pp 285-288

[3] A. Kain and M. Macon, "Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction", In Proceedings of ICASSP '01, May 2001.

[4] A. Kain and M. Macon, "Text-to-speech voice adaptation from sparse training data", In Proceedings of ICSLP '98, vol. 7, pp. 2847-2850.

[5] Alexander Kain. "High Resolution Voice Transformation", PhD thesis, OGI School of Science and Engineering at Oregon Health and Science University, 2001.

[6] A. Schmidt-Neilsen and D.P. Brock, "Speaker recognizability testing for voice coders", In Proceedings of ICASSP '96, vol. 2, pp. 1149-1152.

[7] M. Abe, K. Shikano and H. Kuwabara, "Cross-language voice conversion", In proceedings of ICASSP '90, pp. 345-348.

[8] Masanobu Abe, Satoshi Nakamura, Kiyohiro Shikano, and Hisao Kuwabara, "Voice conversion through vector quantization", In Proceedings of the ICASSP, pp. 655-658., 1988.

[9] Ning Bi and Yingyong Qi, "Application of Speech Conversion to Alaryngeal Speech Enhancement", IEEE Trans. on Speech and Audio Processing, vol. 5, pp. 97-105, March 1997.

[10] J. Hosom, A. Kain, T. Mishra, J. van Santen, M. Fried-Oken, and J. Staehely, "Intelligibility of modifications to dysarthric speech", In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, May 2003.

[11] John Kominek and Alan W. Black, "CMU arctic databases for speech synthesis", Technical Report CMU-LTI-03-177, Carnegie Mellon University Language Technologies Institute, 2003.

[12] J.L. Flanagan, "Speech Analysis, Synthesis and Perception", 2nd ed., Springer-Verlag, 1972.

[13] Thomas F. Quatieri, "Discrete-Time Speech Signal Processing", Prentice Hall, Inc., 2002.

[14] H. Kubawara and Y. Sagisaka, "Acoustic characteristics of speaker individuality: control and conversion", Speech Communication, vol. 16, pp. 165-173, Feb. 1995.

[15] D.H. Klatt and L.C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers", Journal of the Acoustical Society of America, vol. 87, pp. 820-857, Feb. 1990.

[16] John Makhoul, "Linear prediction: A tutorial review", Proceedings of the IEEE, vol. 63, pp. 561-580, April 1975.

[17] Lawrence Rabiner and B.H. Juang, "Fundamentals of Speech Recognition", Pearson Education, Inc., 1993.

[18] Norman Levinson, "The weiner rms (root mean square) error criterion in filter design and prediction", Journal of Mathematical Physics, vol. 25, pp. 261-278, 1947.

[19] J. Durbin, "Efficient estimation of parameters in moving-average models", Biometrika, vol. 46, pp. 306-316, 1959.

[20] F. Itakura, "Line Spectrum representation of linear predictive coefficients of speech signals", Journal of the Acoustic Society of America, vol. 57, pp. 537, 1975.

[21] Wai C. Chu, "Speech coding Algorithms", Wiley-Interscience, 2003.

[22] K.K. Paliwal. "Interpolation properties of linear prediction parametric representations", In Proceedings of Eurospeech, Madrid, Spain, 1995.

[23] J.R. Crosmer, "Very Low Bit Rate Speech Coding Using the Line Spectrum Pair Transformation of the LPC Coefficients", PhD thesis, Georgia Institute of Technology, 1985.

[24] Sassan Ahmadi and Andreas S. Spanias, "Cepstrum-Based Pitch Detection Using a New Statistical V/UV Classification Algorithm", IEEE Trans. on speech and audio processing, vol. 7, pp. 333-338, May 1999.

[25] F. Soong and B.H. Juang, "Line spectrum pair and speech data compression", In Proceedings of the ICASSP, vol. 1, pp. 1.10.1-1.10.4, 1984.

[26] B.S. Atal, L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", Journal of the Acoustic Society of America, vol. 50, pp. 637-55 1971.

[27] Micheal Hart. "Project gutenberg", http://promo.net/pg.

[28] K. Shikano, K.F. Lee, and R. Reddy, "Speaker adaptation through vector quantization", In Proceedings of the ICASSP, pp. 2643-2646, 1986.

[29] P. Zhan and M. Westphal, "Speaker normalisation based on frequency warping", In Proceedings of the ICASSP, pp. 1039-1042, 1997.

[30] Levent M. Arslan, "Speaker transformation algorithm using segmental codebooks (STASC)", Speech Communication, February 1999.

[31] Oytun Turk, "New methods for voice conversion", Masters thesis, Bogaziçi University, 2003.

[32] Oytun Turk and Levent M. Arslan, "Subband-based voice conversion", In Proceedings of the International Conference on Spoken Language Processing, pp. 289-292, 2002.

[33] Christina Orphanidou, "Voice morphing", Masters thesis, Linacre College, University of Oxford, 2001.

[34] Mark Wilde, "Controlling Performance In Voice Conversion With Probabilistic Principal Component Analysis", Master's Thesis, Tulane University, 2004.

[35] Christopher M. Bishop, Marcus Svensen, and C. Williams, "Gtm: The generative topographic mapping", Neural Computation, vol. 10, pp. 215-234, 1998.

[36] Yannis Stylianou, Olivier Cappe, and Eric Moulines, "Statistical methods for voice quality transformation", In Proceedings of Eurospeech, Madrid, Spain, 1995.

[37] Yannis Stylianou, Olivier Cappe, and Eric Moulines, "Continuous probabilistic transform for voice conversion", IEEE Transactions on Speech and Audio Processing, vol. 6, pp. 131-142, March 1998.

[38] Tomoki Toda, Jinlin Lu, Hiroshi Saruwatari, and Kiyohiro Shikano, "Straight-based voice conversion algorithm based on gaussian mixture model", In Proceedings of the International Conference on Spoken Language Processing, 2000.

[39] Hui Ye and Steve young, "Perceptually Weighted linear transformations for voice conversion", Eurospeech, 2003.

[40] Athanasioa Mouchtaris, Jan Van der Spiegel and Paul Mueller, " Non-parallel training for voice conversion by maximum lielihood constrained adaptation", In Proceedings of the ICASSP, 2004.

[41] Douglas A. Reynolds and Richard C. Rose, "Robust Text-Independent speaker identification using gaussian mixture models", IEEE trans. on speech and audio processing, vol. 3, 1995.

[42] Jeff A. Bilmes, "A gentle tutorial of the EM algorithm and it application to parameter estimation for gaussian mixture and hidden markov models", Technical Report, University of Berkeley, ICSI-TR-97-021, 1997.

[43] Nandakishore Kambhatla, "Local models and gaussian mixture models for statistical data processing", Phd thesis, Oregon Graduate Institute of Science and technology, 1996.

[44] T. Watanabe, T. Murakami, and M. Namba, "Transformation of spectral envelope for voice conversion based on radial basis function networks", In Proceedings of the International Conference on Spoken Language Processing, Denver, September 2002.

[45] M. Narendranath, H. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks", Speech Communication, vol. 16, pp. 207-216, 1995.

[46] Ozgul Salor, M. Demirekler, and Bryan Pellom, "A system for voice conversion based on adaptive filtering and line spectral frequency distance optimization for text-to- speech synthesis", In Proceedings of Eurospeech, pp. 2417-2420, September 2003.

[47] Ki Seung Lee, Dae Hee Youn, and Il Whan Cha, "Voice personality transformation using an orthogonal vector space conversion", In Proceedings of Eurospeech, vol. 1, pp. 427-430, 1995.

[48] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers", Neural Computatzon, vol. 11, pp. 443-482, February 1999.

[49] Zoubin Ghahramani and Geoffrey E. Hinton, "The EM algorithm for mixtures of factor analyzers", Technical Report CRG-TR-96-1, University of Toronto, 1996.

[50] G. E. Hinton, P. Dayan, and M. Revow, "Modelling the manifolds of images of hand- written digits", IEEE Trans. on Neural Networks, vol. 8, pp. 65-74, 1997.

[51] Perry Moerland, "A comparison of mixture models for density estimation", In Proceedings of the International Conference on Artificial Neural Networks, 1999.

[52] I.T. Jolliffe, "Principal component analysis", Springer series in statistics, 1986.

[53] Geoffrey MaLachlan and David Peel, "Finite mixture models", Wiley Interscience, 2001.

[54] Harry H. Harman, "Modern factor analysis", The university of Chicago press, 1960.

[55] Stephen J. Roberts and William D. Penny, "Mixtures of Independent Component Analysers", Proceedings of the International Conference on Artificial Neural Networks, pp. 527-534, August 21-25, 2001.

[56] D.J. Miller and H.S. Uyar, "A mixture of experts classfier with learning based on both labelled and unlabelled data", Advances in Neural processing systems, vol. 9, pp. 571-577, 1997.

[57] Zoubin Ghahramani and Matthew J. Beal, "Variational Inference for Bayesian Mixtures of Factor Analysers", Neural Information Processing Systems, vol. 12, pp. 449-455, 2000.